

Freeform Search

Database:

US Pre-Grant Publication Full-Text Database
 US Patents Full-Text Database
 US OCR Full-Text Database
 EPO Abstracts Database
 JPO Abstracts Database
 Derwent World Patents Index
 IBM Technical Disclosure Bulletins

Term:

L4 and filter

Display: Documents in Display Format: Starting with Number Generate: ☐ Hit List ☒ Hit Count ☐ Side by Side ☐ Image

Search

Clear

Interrupt

Search History

DATE: Wednesday, March 17, 2004 [Printable Copy](#) [Create Case](#)

Set Name Query
 side by side

Hit Count Set Name
 result set

DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR

| | | | |
|-----------|--|-----|-----------|
| <u>L5</u> | L4 and filter | 11 | <u>L5</u> |
| <u>L4</u> | (contract near management near system) | 93 | <u>L4</u> |
| <u>L3</u> | L1 and (contract) | 9 | <u>L3</u> |
| <u>L2</u> | L1 and (contract near dataset) | 2 | <u>L2</u> |
| <u>L1</u> | system near application near product\$ | 173 | <u>L1</u> |

END OF SEARCH HISTORY

First Hit**End of Result Set**

Generate Collection

Print

L9: Entry 1 of 1

File: PGPB

Apr 17, 2003

DOCUMENT-IDENTIFIER: US 20030074277 A1

TITLE: Method and apparatus for automatically reviewing application information and preparing responsive communications

Pre-Grant Publication (PGPub) Document Number:
20030074277Detail Description Paragraph:

[0057] Server computer 106 could be, for example, one or more stationary desktop, mainframe, or other computing devices. Server computer 106 receives information from client computers 102 over the network 104, and accesses information, files, and data stored within a database 108. In one embodiment, an application server 110 resident on server 106 can execute scripts, which enable various screens to be displayed on client computers 102. A database server 112 resident on server 106 interacts with application server 110 to access database 108. In addition, a file server 114 resident on server 106 interacts with application server 110 to access files (e.g., HTML files) stored on server 110, database 108, or elsewhere.

Detail Description Paragraph:

[0058] In other embodiments, multiple servers 106 could be included in the system. Each of the multiple servers 106 could include an application server 110, database server 112, and file server 114. Alternatively, the application server 110, database server 112, and/or file server 114 could be implemented on separate server computers 106.

Detail Description Paragraph:

[0063] Database 108 can include one or more types of information storage devices. Some or all of database 108 can be resident on server computer 106, or can reside on one or more stand-alone computers (not shown). In one embodiment, database 108 is used to store user information 120, one or more rule sets 122, application data and status information 124, commission information 126, and other information 128. More or different information could be stored in database 108, in various other embodiments. In addition, the information could be distributed across more than one information storage device.

Detail Description Paragraph:

[0087] Some or all of these features may access information stored within a database 402. In one embodiment, a client requests information within the database through a server.

Detail Description Paragraph:

[0091] Some or all of these features may access information stored within the database 402. In one embodiment, a client requests information within the database through interactions with a server.

Detail Description Paragraph:

[0101] In block 706, the user submits the information in the first application information input page (e.g., by clicking a "Submit" or "Next" page element). The

server then makes a determination whether all submitted inputs are valid, in block 708. For example, the server could check the database to see whether a bid (e.g., a quote) already exists for the Potential Customer. In addition, the server could check portions of the submitted information to make sure the information is valid (e.g., the server could search to determine whether the tax ID matches the customer name). If the information is not valid, then the server causes the client computer to display an error message. The user is then given the opportunity to correct the information, in one embodiment.

Detail Description Paragraph:

[0103] The second application information input page prompts the user to enter information relating to the desired coverage. For example, since Workers' Compensation coverage differs from state to state, a user could enter information into a Workers' Compensation class code table, with one or more rows of information for each state in which coverage is desired. In each row, the user could enter: the state, the class code for the type of insurance; the category of Workers' Compensation (e.g., gas or lease operator, domestic workers inside, etc.); the number of employees to be covered under the class code; the approximate annual payroll associated with the class code; and the rate. In addition, the user could be prompted to enter additional information. For example, for each state, the user could enter the EMod, discount, credit, and SUTA rate information. In an alternate embodiment, the system could auto-fill some of these fields, by looking up the various information for each state in one or more tables (e.g., class code and/or SUTA rate tables) available to the server and/or stored in the database.

Detail Description Paragraph:

[0109] The application information is then stored, in block 718. In one embodiment, the information is stored in a database (e.g., database 108, FIG. 1), accessible to the system. The information can also be incorporated into a form, in one embodiment, which includes the Agent information, the Potential Customer information, and other information. The form is also or alternatively stored, in other embodiments. The server then causes the client computer to display an "Application Submittal Successful" screen, in block 720, and the process ends.

Detail Description Paragraph:

[0132] If a user modifies any entries within the editable fields 1204, 1206, 1208 and selects the "Save" option, then the application information in the database will be updated. In one embodiment, any previously submitted application information could be modified. For example, a user could add or delete a Workers' Compensation code, or edit any other application information. Similarly, if the user selects the "Delete" option, the application information in the database will be removed.

Detail Description Paragraph:

[0137] If the user selects the "View Quote" or "View Contract" options, then the quote or contract information is retrieved from the database, and the server causes the client to display the quote or contract.

Detail Description Paragraph:

[0138] If the user selects the "Reject" option, then the server causes the client to prompt the user to enter a reason for the rejection. The rejection information is stored in the database, and the status of the application is changed to "Rejected."

Detail Description Paragraph:

[0143] The method begins, in block 1302, when a registered user indicates (e.g., by clicking the "Change My Info" menu item), that the user wants to change the user's information that is stored in the system database. In one embodiment, the client requests a user information page from the server, and the server causes the user information page to be displayed on the client computer, in block 1304.

Detail Description Paragraph:

[0148] If the request is accepted, then the user information is modified in the database, in block 1318. In one embodiment, the user is then notified, in block 1320, and the method ends. In one embodiment, the user is sent an e-mail, indicating that their request to change their user information was accepted. In another embodiment, the server causes the client computer to display a message with the same indication.

Detail Description Paragraph:

[0150] FIG. 14 illustrates a flowchart of a method for looking up information in accordance with an embodiment of the invention. The method begins, in block 1402, when a registered user indicates (e.g., by clicking one of the "Look Up Class Codes," "Look Up Code Cost" or "Look Up SUTA" menu items), that the user wants to view information that is stored in the system database. In one embodiment, the client requests an information look up page (i.e., a search page) from the server, and the server causes the information look up page to be displayed on the client computer, in block 1404.

Detail Description Paragraph:

[0152] Based on the submitted criteria, then in block 1408, the server performs a search of appropriate tables of information stored in the database. For example, the server could search a Workers' Compensation class code and description table, a SUTA rate table, or other stored information. A determination is made, in block 1410, whether information was found that matched the submitted look up criteria. If so, then the server causes the client to display the requested information, in block 1412. If not, then the server causes the client to display an error message, in block 1414. The method then ends.

Detail Description Paragraph:

[0155] FIG. 15 illustrates a flowchart of a method for viewing commission information in accordance with an embodiment of the invention. The method begins, in block 1502, when a registered user indicates (e.g., by clicking the "View Commissions" menu item), that the user wants to view commission information that is stored in the system database. In one embodiment, the client requests a commission selection page from the server, and the server causes the commission selection page to be displayed on the client computer, in block 1504.

Detail Description Paragraph:

[0160] FIG. 16 illustrates a flowchart of a method for an administration-level user to view or change user information in accordance with an embodiment of the invention. The method begins, in block 1602, when an administration-level user indicates (e.g., by clicking one of the "Change User Info" or "View New User" menu items), that the user wants to view or change user information that is stored in the system database.

Detail Description Paragraph:

[0163] In one embodiment, along with other options provided along with the user information page, the administration-level user is given the options to delete the selected user's information, update the information, and print a user agreement. A determination is made, in block 1610, whether the administration-level user has selected the delete option. If so, then the database is updated, in block 1614, by deleting the corresponding user information.

Detail Description Paragraph:

[0164] If the option to delete the user has not been selected, a determination is made, in block 1612, whether the administration-level user has selected the update information option. Generally, a user would select this option after having modified one or more fields of user information. If the user has selected the update information option, then the database is updated with the modified

information, in block 1614.

Detail Description Paragraph:

[0172] The method begins, in block 1802, when an administration-level user indicates (e.g., by clicking the "Generate Report" menu item), that the user wants to generate a report from information that is stored in the system database. In one embodiment, the client requests a report selection page from the server, and the server causes the report selection page to be displayed on the client computer, in block 1804.

Detail Description Paragraph:

[0175] The user inputs the desired search criteria, in block 1810. The server then collects the corresponding report information, in block 1812, from the database. Using the collected information, the server generates a report, in block 1814, and the method ends. In one embodiment, the server causes the client to display the report on the screen. In another embodiment, the server can notify the user of a location of the report, or provide a link to the report. The user can then access and/or print the report.

First Hit

Generate Collection

Print

L3: Entry 2 of 9

File: PGPB

Apr 17, 2003

DOCUMENT-IDENTIFIER: US 20030074277 A1

TITLE: Method and apparatus for automatically reviewing application information and preparing responsive communications

Summary of Invention Paragraph:

[0004] If insurance will be offered, a representative calculates a premium payment, and prepares a quote for the individual's review. If the quote appears accurate and acceptable to the individual, a representative prepares a contract. Once signed, the contract is binding, and coverage will be in force as of a specified coverage date.

Detail Description Paragraph:

[0040] "Application Review Communication" or "ARC" mean a document, e-mail, letter or other communication, in electronic form, which includes a bid, a quote, an offer, a contract, an indication of an application's acceptance or rejection, or another written communication produced in response to an application review task.

Detail Description Paragraph:

[0047] Prior art ways of evaluating insurance and other types of applications predominantly involve human efforts. In particular, prior art underwriting processes are performed by one or more individuals who manually review the application, apply underwriting criteria, make decisions regarding whether or not to offer coverage, and initiate quote or contract production. Prior art methods of evaluating other types of applications are similar, in regards to the level of human involvement in the process.

Detail Description Paragraph:

[0076] A determination is made, in block 208, whether the review process resulted in a decision to automatically accept the Application. If so, then an acceptance notification process is performed, in block 210, and the method ends. In one embodiment, this involves generating and storing information for an ARC. For example, but not by way of limitation, the ARC could be an insurance quote or contract. In one embodiment, which will be described in more detail in conjunction with FIG. 8 (block 818), an intermediate, manual review is performed before sending the Requester the ARC.

Detail Description Paragraph:

[0125] If one or more applications do match the selected type or criteria, then in block 912, a determination is made whether an ARC exists for one or more of the matching applications. For example, if an application is an insurance application, an ARC could be a quote or a contract based off of the application. If an ARC exists for one or more of the matching applications, then in one embodiment, a link to each ARC is displayed on the client computer, and/or a link to each ARC is e-mailed to the user, in block 914.

Detail Description Paragraph:

[0131] Referring back to FIG. 12, each of the selectable options in list 1202 will be briefly described below. In one embodiment, the user is given the ability to choose from one or more of the following options: save edits to the application form; initiate a quoting process; requote a previously quoted application; get loss

information; perform a new client procedure; perform a client renewal procedure; get a manual approval of an application or ARC; refer an application; view a quote; view a contract; reject the application; delete the application; perform some calculations; or generate an ARC. Other options also could be provided to the user. For example, when the application pertains to Workers' Compensation insurance, the user could be given the opportunity to select an option that will determine evidence of Workers' Compensation insurance, or to create a Workers' Compensation certificate. In other embodiments, more, fewer or different user options could be provided.

Detail Description Paragraph:

[0137] If the user selects the "View Quote" or "View Contract" options, then the quote or contract information is retrieved from the database, and the server causes the client to display the quote or contract.

CLAIMS:

2. The method of claim 1, wherein the product is an insurance product, the one or more electronically-stored rule sets are one or more sets of underwriting criteria, and the ARC is an insurance quote or contract.
9. The method of claim 8, wherein the application is an application for insurance coverage, the product is an insurance product, the one or more electronically-stored rule sets are one or more sets of underwriting criteria, and the ARC is an insurance quote or contract.
10. The method of claim 8, further comprising: the first computer sending a request to the second computer to access an online product application system; the second computer causing the first computer to display a first login screen which prompts the user of the first computer for login information; the first computer sending the login information to the second computer, where the login information identifies the user of the first computer; and the second computer determining whether the login information is valid.
14. The method of claim 11, further comprising: if the status of the previously submitted application indicates that a contract exists for the previously submitted application, the second computer causing the first computer to display information enabling the user to access the contract.
15. The method of claim 14, wherein causing the first computer to display information comprises: the second computer causing the first computer to display a selectable link identifying the contract; and if the user selects the selectable link, the second computer sending an electronic version of the contract to the first computer.
28. A method performed by one or more computers, the method comprising: maintaining, in electronic form, one or more sets of user information for one or more users of an online product application system, wherein the online product application system enables a first computer to send application information to a second computer, where the application information includes information pertaining to a potential customer of a provider of a product, and the online product application system also causes the second computer to decide, based on the application information and one or more electronically-stored rule sets, whether or not to generate an application review communication (ARC) for the potential customer, which describes terms under which the product is offered, and if the second computer decides to generate the ARC, the online product application system causes the second computer to automatically generate and store the ARC; and maintaining, in electronic form, the application information, the one or more rule sets, and the ARC.

29. The method of claim 28, wherein the product is an insurance product, the one or more electronically-stored rule sets are one or more sets of underwriting criteria, and the ARC is an insurance quote or contract.

37. The method of claim 36, wherein the product is an insurance product, the one or more electronically-stored rule sets are one or more sets of underwriting criteria, and the ARC is an insurance quote or contract.

42. The method of claim 41, wherein the product is an insurance product, the one or more electronically-stored rule sets are one or more sets of underwriting criteria, and the ARC is an insurance quote or contract.

49. The first computer of claim 48, wherein the product is an insurance product, the one or more electronically-stored rule sets are one or more sets of underwriting criteria, and the ARC is an insurance quote or contract.

54. The second computer of claim 53, wherein the product is an insurance product, the one or more electronically-stored rule sets are one or more sets of underwriting criteria, and the ARC is an insurance quote or contract.

First Hit

Generate Collection

Print

L3: Entry 3 of 9

File: PGPB

Feb 27, 2003

DOCUMENT-IDENTIFIER: US 20030041049 A1

TITLE: Management of contract dataAbstract Paragraph:

A system and method for managing contract data. A contract dataset is received by a decentralized execution system (DES) from a procurement contract management system (PCMS). The contract dataset is passed through a software filter that determines whether to store the contract dataset or a portion thereof in a relational database that includes contract datasets, vendor datasets, and purchase item datasets. If the software filter determines not to so store the contract dataset or the portion thereof, then the software filter determines whether to store the contract dataset or a portion thereof in a special database of the DES. An execution document at the DES is updated by replacing an existing attribute value of the execution document by a new attribute value communicated to the DES by the PCMS. Additionally, a contract is archived if each such DES sends permission to the PCMS to archive the contract.

Summary of Invention Paragraph:

[0002] The present invention relates to a system and method for managing contract data that is transferred between discrete contract management systems.

Summary of Invention Paragraph:

[0004] An online financial software package known as Systems Applications and Products (SAP) includes software that can be used for managing contract data, but is inefficient for managing contract data that is transferred between discrete SAP systems. Accordingly, there is a need for an efficient system and method for managing contract data that is transferred between discrete SAP systems.

Summary of Invention Paragraph:

[0005] The present invention provides a method for managing contract data, comprising:

Summary of Invention Paragraph:

[0006] receiving a contract dataset by a decentralized execution system (DES) from a procurement contract management system (PCMS); and

Summary of Invention Paragraph:

[0007] passing the contract dataset through a software filter that determines whether to store the contract dataset or a first portion thereof in a relational database of the DES, said relational database including contract datasets, vendor datasets, and purchase item datasets.

Summary of Invention Paragraph:

[0008] The present invention provides a method for updating an execution document relating to a contract, said method comprising:

Summary of Invention Paragraph:

[0009] having an execution document at a decentralized execution system (DES) of a procurement contract management system (PCMS), said execution document being derived from a contract dataset in the DES, said execution document having an

existing attribute value for a purchase item in the contract dataset;

Summary of Invention Paragraph:

[0012] The present invention provides a method of contract archiving, comprising:

Summary of Invention Paragraph:

[0013] sending a list of I identifiers by a procurement contract management system (PCMS) to at least one decentralized execution system (DES), said I at least 1, each identifier of the I identifiers identifying a contract dataset in the PCMS earmarked by the PCMS for archiving;

Summary of Invention Paragraph:

[0014] receiving by the PCMS a return list of M of the I identifiers from each DES of the at least one DES in response to said sending, said M in a range of 0.ltoreq.M.ltoreq.I, said return list being DES-specific, each said contract dataset identified in the return list of each DES having been approved by said each DES for archiving; and

Summary of Invention Paragraph:

[0015] archiving by the PCMS each contract dataset identified in the list of I identifiers and appearing in an intersection list of the return lists, if the intersection list is not empty.

Summary of Invention Paragraph:

[0016] The present invention provides a system for managing contract data, comprising software at a decentralized execution system (DES), said software adapted to:

Summary of Invention Paragraph:

[0017] receive a contract dataset by the DES from a procurement contract management system (PCMS); and

Summary of Invention Paragraph:

[0018] pass the contract dataset through a software filter that is adapted to determine whether to store the contract dataset or a first portion thereof in a relational database of the DES, said relational database adapted to include contract datasets, vendor datasets, and purchase item datasets.

Summary of Invention Paragraph:

[0019] The present invention provides a system for updating an execution document relating to a contract, comprising a decentralized execution system (DES) of a procurement contract management system (PCMS), said DES having software adapted:

Summary of Invention Paragraph:

[0020] to have an execution document at the DES, said execution document being derived from a contract dataset in the DES, said execution document having an existing attribute value for a purchase item in the contract dataset;

Summary of Invention Paragraph:

[0023] The present invention provides a system for contract archiving, comprising a procurement contract management system (PCMS) having software adapted:

Summary of Invention Paragraph:

[0024] to send a list of I identifiers to at least one decentralized execution system (DES), said I at least 1, each identifier of the I identifiers identifying a contract dataset in the PCMS earmarked by the PCMS for archiving;

Summary of Invention Paragraph:

[0025] to receive a return list of M of the I identifiers from each DES of the at least one DES in response to having sent the list of I identifiers to each said

DES, said M in a range of 0.1to req.M.1to req.I, said return list being DES-specific, each said contract dataset identified in the return list of each DES having been approved by said each DES for archiving; and

Summary of Invention Paragraph:

[0026] to archive each contract dataset identified in the list of I identifiers and appearing in an intersection list of the return lists, if the intersection list is not empty.

Summary of Invention Paragraph:

[0027] The present invention provides an efficient system and method for managing contract data that is transferred between discrete SAP systems. The present invention also provides an automated and efficient system and method for contract archiving.

Brief Description of Drawings Paragraph:

[0028] FIG. 1 is a block diagram of a contract management architecture that includes a decentralized execution system (DES) coupled to a procurement contract management system (PCMS), in accordance with embodiments of the present invention.

Brief Description of Drawings Paragraph:

[0029] FIG. 2 depicts relationships between a contract, a contract dataset, a contract deltatadataset, and an execution document, in accordance with embodiments of the present invention.

Brief Description of Drawings Paragraph:

[0031] FIG. 4 depicts a layout of the contract dataset of FIG. 2, in accordance with embodiments of the present invention.

Brief Description of Drawings Paragraph:

[0034] FIG. 7 is a flow chart for processing a new or updated contract dataset, in accordance with embodiments of the present invention.

Brief Description of Drawings Paragraph:

[0036] FIG. 9 depicts archiving a contract, in accordance with embodiments of the present invention.

Brief Description of Drawings Paragraph:

[0037] FIG. 10 is a block diagram of a computer configuration for the contract management architecture of FIG. 1, in accordance with embodiments of the present invention.

Detail Description Paragraph:

[0038] FIG. 1 is a block diagram of a contract management architecture 10 that includes a decentralized execution system (DES) 14 coupled to a procurement contract management system (PCMS) 12, in accordance with embodiments of the present invention. The PCMS 12 and the DES 14 may each independently be a Systems Applications and Products (SAP) system or a non-SAP system. Definitionally, a SAP system functions by executing SAP software.

Detail Description Paragraph:

[0039] The contract management architecture 10 manages contracts for the sale of goods (e.g., materials, devices, machines, vehicles, etc.) and services (e.g., service to repair, install, fabricate, advertise, etc.) for use by buyers of the goods and services. Such goods and services are called "purchase items." A contracts database 16 stores the contracts in their exact wording, while the PCMS 12 receives from the contracts database 16 selected information (e.g., vendor, purchase item(s), price, payment terms, termination date, etc.) relating to some or all of the contracts stored in the contracts database 16. The contract data stored in the PCMS 12 for each contract is called a "contract dataset." A "dataset" is

defined herein as a collection of data, such as a database, one or more files of data, one or more tables of data, etc. A data path 13 between the contracts database 16 and the PCMS 12 may be electronic or manual.

Detail Description Paragraph:

[0040] The PCMS 12 serves as a master repository for contract data and feeds such contract data, as contracts are created and updated, to one or more DES such as the DES 14. Each DES serves to execute the functionality of selected contracts, such as to create and update execution documents (e.g., purchase orders, scheduling agreements, etc.) relating to the selected contracts. Thus the DES 14 requires data for those contracts that are active (i.e., being used or intended to be used) in the DES 14. The PCMS 12 feeds contract datasets to the DES 14 for such contracts that are active in the DES 14, but also for contracts that are not active in the DES 14. Since the DES 14 needs contract datasets only for its active contracts, the DES 14 selectively filters contract datasets received from the PCMS 12 and stores in its main relational database only contract data for its active contracts, as will be described infra. A data path 15 between the purchase the PCMS 12 and the DES 14 is electronic and computer automated. The data path 15 might represent a data communications network between PCMS 12 at a the central site and the DES 14 at a remote site. Definitionally, a data communications network comprises communication lines over which data is transmitted from one node to another, and each said node may include, inter alia, a computer, a terminal, a communication control unit, etc.

Detail Description Paragraph:

[0041] Each contract states a vendor (i.e., seller) and purchase items to be purchased by a named purchaser. Accordingly, the contract management architecture 10 includes a vendor database 18 and a purchase item database 20. The vendor database 18 is a master repository of vendors and stores, in vendor datasets, information about each vendor such as identification (e.g., a vendor number), name of vendor, address, telephone number, etc. The PCMS 12 receives from the vendor database 18 vendor information (i.e., vendor datasets) that pertain to the contract datasets stored within the PCMS 12. A data path 21 between the vendor database 18 and the PCMS 12 may be electronic or manual. The DES 14 receives from the vendor database 18 vendor information (i.e., vendor datasets) that relate to contracts that are active or may become active in the DES 14. A data path 22 between the vendor database 18 and the DES 14 may be electronic or manual.

Detail Description Paragraph:

[0042] The purchase item database 20 is a master repository of purchase items and stores, in purchase item datasets, information about each purchase item such as identification (e.g., a purchase item number) and characteristics (e.g., size, weight, color), descriptive text, etc. The PCMS 12 receives from the purchase item database 20 purchase item information (i.e., purchase item datasets) that pertain to the contract datasets stored within the PCMS 12. A data path 23 between the purchase item database 20 and the PCMS 12 may be electronic or manual. The DES 14 receives from the purchase item database 20 purchase item information (i.e., purchase item datasets) that relate to contracts that are active or may become active in the DES 14. A data path 24 between the purchase item database 20 and the DES 14 may be electronic or manual.

Detail Description Paragraph:

[0043] FIG. 2 depicts relationships between a contract 26, a contract dataset 28, a contract deltadataset 30, and an execution document 32, in accordance with embodiments of the present invention. A contract 26 for the sale of purchase items (i.e., goods or services), as used herein, is a legally binding agreement, in writing, between a purchaser and a vendor of the purchase items. The contract 26 consists of all of the words of the agreement. A contract dataset 28 is a collection of data comprising terms (e.g., vendor, purchaser, purchase item(s), price, payment terms, termination date, etc.) of the contract. The contract

deltadataset 30 is a collection of data that updates an already existing contract dataset. The contract deltadataset 30 may include such information as added purchase items to an existing contract, a change in price or a new price of a purchase item in an existing contract, changes in delivery terms such as free on board (F.O.B.), free alongside (F.A.S.), change in payment terms, etc. The present invention processes added purchase items in the contract deltadataset 30 as discussed infra in accordance with FIG. 7. The present invention processes other changes such as new or changed prices of the contract deltadataset 30 as discussed infra in accordance with FIG. 8. The execution document 32 includes, inter alia, a purchase order, a scheduling agreement, etc. As seen in FIG. 2, the contract dataset 28 is derived from the contract 26 and is said to be keyed to the contract 26. The contract deltadataset 30 feeds the contract dataset 28 and is said to be keyed to the contract dataset 28. The execution document 32 is derived from the contract dataset 28.

Detail Description Paragraph:

[0044] FIG. 3 depicts a layout of a relational database 40 of the DES 14 of FIG. 1, in accordance with embodiments of the present invention. In FIG. 3, the relational database 40 comprises contract datasets 42, vendor datasets 44, purchase item datasets 46, and execution documents, 48. If the DES 14 is a SAP system, then the relational database 40 is a SAP relational database, and if the DES 14 is a non-SAP system, then the relational database 40 is a non-SAP relational database. Definitionally, a SAP relational database is a relational database that functions under control of SAP software.

Detail Description Paragraph:

[0045] FIG. 4 depicts a layout of a contract dataset 50, such as the contract dataset 28 of FIG. 2, in accordance with embodiments of the present invention. In FIG. 4, the contract dataset 50 comprises a contract identification 52 (e.g., contract identification number), a vendor identification 54 (e.g., vendor identification number), purchase item(s) 56, and other contract terms or data 58.

Detail Description Paragraph:

[0046] Purchase orders and scheduling agreements are examples of execution documents. FIG. 5 depicts entries that may appear in a purchase order 60, in accordance with embodiments of the present invention. The price in a purchase order applies through the term (i.e., time period) of the contract. If the price is changed in accordance with a new or renewed contract (or for any other reason), the purchase order will be modified to incorporate the price change as described infra in conjunction with FIG. 8. FIG. 6 depicts entries that may appear in a scheduling agreement 62, in accordance with embodiments of the present invention. A scheduling agreement includes a schedule for delivering the purchase items bargained for by the purchaser. If the price is determined by a price in effect at the time of delivery, then the scheduling agreement will be updated to reflect any change in price that occurs prior to delivery as described infra in conjunction with FIG. 8.

Detail Description Paragraph:

[0047] FIG. 7 is a flow chart for DES software 65 (called "DES FILTER" software) that processes a "contract datagroup" received by the DES 14 from the PCMS 12 of FIG. 1, as denoted in block 64 and in accordance with embodiments of the present invention. A "contract datagroup" is defined herein as being either a contract dataset or a contract deltadataset having new or changed purchase items. The DES software 65 also processes a new purchase item that is added to a relational database (RDBS) of the DES 14 as denoted in block 78. Although the DES FILTER software of the present invention does not currently exist in SAP, the scope of the present invention includes the DES FILTER software as either SAP software or non-SAP software. In relation to use of the DES FILTER software, the scope of the present invention includes the PCMS 12 and the DES 14 as each independently being a SAP system or a non-SAP system.

Detail Description Paragraph:

[0048] The DES software 65 is applicable to, inter alia, the following situation. If the PCMS 12 and the DES 14 of FIG. 1 are each a SAP system, then the PCMS 12 pushes all contract datagroups in its database into each such DES system to which the PCMS 12 is coupled. However, the DES 14 does not process or execute all contract datasets that exist in the PCMS 12, but only those contract datasets that are active in the DES 14. Thus, it would be inefficient and wasteful for the DES 14 to accept and store in its relational database 40 (see FIG. 3) all contract datagroups (or contents thereof) that the DES 14 receives from the PCMS 12. Accordingly, the DES software 65 selectively processes (i.e., filters) the datagroups received from the PCMS 12.

Detail Description Paragraph:

[0049] As stated supra, the DES 14 receives a contract datagroup D.sub.G from the PCMS 12 as indicated in block 64. The contract datagroup D.sub.G is either a contract dataset or a contract deltadataset with one or more new purchase items. The contract datagroup D.sub.G identifies N purchase items ($N \geq 1$) that are purchasable from a vendor V keyed to the contract datagroup D.sub.G (i.e., identified in the contract as a vendor). If the contract datagroup D.sub.G is the contract dataset, then the contract datagroup D.sub.G identifies the vendor V. If the contract datagroup D.sub.G is the contract deltadataset, then the contract datagroup D.sub.G does not have to identify the vendor V, since the vendor V has been previously identified to the already-existing contract dataset. The DES 14 comprises the relational database 40 of FIG. 3 which includes the contract datasets 42, the vendor datasets 44 having vendors, and purchase item datasets 46 having purchase item(s). The relational database 40 may also include execution documents 48.

Detail Description Paragraph:

[0050] In FIG. 7, decision block 66 determines which, if any, of the N purchase items identified in the contract datagroup D.sub.G matches a purchase item in the purchase item datasets 46 stored in the DES relational database (DES RDBS) 40 (see FIG. 3). The decision block 66 also determines a total number K of such purchase items in D.sub.G that do not so match a purchase item in the purchase item datasets 46 of the DES RDBS 40 of FIG. 3. K is in a range of $0 \leq K \leq N$.

Detail Description Paragraph:

[0051] If $K < N$ then a remaining $N - K$ purchase items in D.sub.G are in the DES RDBS 40 and for the remaining $N - K$ purchase items, the subsequent processing depends on whether the contract datagroup D.sub.G is the contract dataset or the contract deltadataset. If the contract datagroup D.sub.G is the contract dataset, then the decision block 68 determines whether the vendor V matches a vendor in the vendor datasets 44 (see FIG. 4). If the vendor V so matches a vendor in the vendor datasets 44, then block 69 adds a subset of D.sub.G to the contract datasets 42 of the relational database 40. This subset of D.sub.G is the remaining $N - K$ purchase items in D.sub.G formed by excluding the K purchase items from D.sub.G. For $K > 0$, the K purchase items are not stored in the relational database 40, since the K purchase items do not exist in the purchase item datasets 46 (see FIG. 3), as discussed supra. If $K > 0$ then the K purchase items not in the purchase item datasets 46 may be stored in a special database of the DES 14, as denoted in block 67. The special database stores contract datasets having one or more purchase items not currently present in the purchase item datasets 46. The contract datasets stored in the special database may be subsequently used to update the contract datasets 42 of the relational database 40 when a new matching purchase item is added in the future to the purchase item datasets 46, as will be explained infra in conjunction with block 78. Although the special database of the present invention does not currently exist in SAP, the scope of the present invention includes the special database as either a SAP database or a non-SAP database.

Detail Description Paragraph:

[0052] Returning to the decision block 66 for the case of $K < N$, the alternative situation of the contract datagroup D.sub.G being the contract deltadataset will now be considered. The contract deltadataset includes N-K purchase items that exist in the purchase items database 46 in relation to a contract dataset D.sub.1, wherein D.sub.1 currently exists the contract datasets 42 (see FIG. 3). Thus, the contract deltadataset is said to be keyed to D.sub.1. Since D.sub.1 is a pre-existing contract dataset with an already-identified vendor, the decision block 68 is bypassed and the block 69 is executed next, which adds to D.sub.1 the remaining N-K purchase items of the contract datagroup D.sub.G. If $K > 0$, then the K purchase items not in the purchase item datasets 46 may be stored in the special database of the DES 14, as denoted in the block 67 as follows. If D.sub.G is keyed to a contract dataset D' in the special database (i.e., if D.sub.G has a contact identification that matches the contract identification of the contract dataset D' in the special database), then the K purchase items of D.sub.G are added to D'. If D.sub.G is not keyed to any contract dataset in the special database, then a new contract dataset D.sub.C1 is formed from D.sub.G such that D.sub.C1 includes the K purchase items of D.sub.G and excludes the remaining N-K purchase items of D.sub.G, and D.sub.C1 is then added to the special database.

Detail Description Paragraph:

[0053] Returning to the decision block 66, the case of $K = N$ is now considered. If $K = N$, then no purchase item in the contract datagroup D.sub.G matches a purchase item in the purchase item datasets 46 stored in the DES RDBS 40 (see FIG. 3). Thus, no portion of the contract datagroup D.sub.G is added to the DES RDBS 40, since none of the purchase item in D.sub.G exist in the DES 14. Instead, the K purchase items not in the purchase item datasets 46 may be stored in the special database of the DES 14, as denoted in the block 67 as follows. If D.sub.G is keyed to a contract dataset D" in the special database, then the N purchase items of D.sub.G are added to D". If D.sub.G is not keyed to any contract dataset in the special database, then a new contract dataset D.sub.C2 is formed from D.sub.G such that D.sub.C2 includes the N purchase items of D.sub.G, and D.sub.C2 is then added to the special database.

Detail Description Paragraph:

[0054] Returning to block 68 (which is pertinent only if the contract datagroup D.sub.G is the contract dataset and is not pertinent if the contract datagroup D.sub.G is the contract deltadataset), if the vendor V does not match a vendor in the vendor datasets 44 of the DES RDBS 40 (see FIG. 3) then a vendor dataset D.sub.V may be added to the vendor datasets 44 of the DES RDBS 40 (see FIG. 3) when a contract based on one or more of the remaining N-K purchase items of D.sub.G is required at the DES 14 (i.e., required due to a need to purchase the one or more of the remaining N-K purchase items of D.sub.G at the DES 14), wherein the vendor dataset D.sub.V is keyed to the vendor V (i.e., includes the vendor V). A manner of adding D.sub.V to the vendor datasets 44 is shown in blocks 70-73 of FIG. 7. In block 70, a DES buyer is sent a message relating to adding D.sub.V to the vendor datasets 44 of the RDBS 40. The DES buyer is keyed to (i.e., authorized to purchase) at least one purchase item of the remaining N-K purchase items. As shown in decision block 71, the DES buyer queries whether the contract based on one or more of the remaining N-K purchase items of D.sub.G is required at the DES 14. If the answer to the query is YES, then the DES buyer may cause D.sub.V to be added to the vendor V datasets 44 by extracting the vendor V from the vendor database 18 (see FIG. 1) as indicated in block 72, followed by adding the vendor V to the vendor datasets 44 as indicated in block 73. If the answer to the query is NO, then the DES buyer waits and requests the vendor V at a time in the future when a contract based on one or more of the remaining N-K purchase items of D.sub.G becomes required at the DES 14 (see block 74), followed by execution of blocks 72 and 73 described supra. Although blocks 70-73 describe a process for adding D.sub.V to the vendor datasets 44, any method that would be obvious to one of ordinary skill in the art may be used for adding D.sub.V to the vendor datasets 44. After the vendor V is added to the vendor datasets 44 as indicated in block 73, the

subset of D.sub.G comprising the remaining N-K purchase items in D.sub.G is added to the contract datasets 42 in the RDBS 40 as indicated in block 69.

Detail Description Paragraph:

[0055] Block 78 of FIG. 7 illustrates how the DES software 65 processes a new purchase item that has been added to the purchase item datasets 46 of the DES RDBS 40 (see FIG. 3). The new purchase item may be derived from the purchase item database 20 of FIG. 1. As discussed supra in conjunction with block 67 of FIG. 7, the special database of the DES 14 (see FIG. 1) may be used to store contract datasets for purchase items received (see block 64) from the PCMS 12 (see

Detail Description Paragraph:

[0056] FIG. 1) wherein such purchase items are not in the purchase item datasets 46. Thus, if a new purchase item is subsequently added to the purchase item datasets 46, as indicated in block 78, then block 76 asks whether the new purchase item exists in a contract dataset D.sub.CS of the special database. If the new purchase item does not exist in a contract dataset D.sub.CS of the special database, then processing for the new purchase item ceases as indicated in block 77.

Detail Description Paragraph:

[0057] However, if in response to the query in block 76, D.sub.CS exists such that the new purchase item is identified in D.sub.CS, and if D.sub.CS identifies a total of J purchase items (J.ltoreq.1), then block 68 determines whether a vendor identified in D.sub.CS matches a vendor in the vendor datasets 44 (see FIG. 3). If the vendor identified in D.sub.CS so matches a vendor in the vendor datasets 44 then the contract datasets 42 (see FIG. 3) are updated with purchase item data in D.sub.CS as follows. If a contract identifier of D.sub.CS matches a contract identifier of a contract dataset D.sub.CR in the relational database 40 (see FIG. 3) then the new purchase item is added to the contract dataset D.sub.CR. However, if the contract identifier of D.sub.CS does not matches a contract identifier of any contract dataset in the relational database 40 then a subset of D.sub.CS is added to the relational database, wherein the subset of D.sub.CS includes the new purchase item. Additionally, if J=1 then D.sub.CS is deleted from the special database, but if J>1 then the new purchase item is deleted from D.sub.CS.

Detail Description Paragraph:

[0058] Returning to the path of blocks 78, 76, and 68, if the vendor identified in D.sub.CS does not match any vendor in the vendor datasets 44 (see FIG. 3), then the new purchase item is further processed in the same manner as was described supra for a purchase item in a contract dataset introduced into the DES 14 via the path of blocks 64 and 68, and is thus further processed in accordance with blocks 70-73, and 69 as described supra.

Detail Description Paragraph:

[0059] In summary, a contract dataset sent to the DES 14 by the PCMS 12 (see FIG. 1) as shown in block 64 of FIG. 7 passes through a software filter provided by the "DES FILTER" software. The filter functionality provided in blocks 66 and 68. Additional filter functionality is provided in block 76 for new purchase items. The software filter functionality determines whether to store the contract dataset being filtered (or a first portion thereof) in the relational database 40 (see FIG. 3) of the DES 14. The software filter functionality further determines whether to store the contract dataset being filtered (or a portion thereof) in the special database of the DES 14. In an embodiment of the software filter functionality, the DES 14 is a first SAP contract management system, the PCMS 12 is a second SAP contract management system, the relational database 40 is a SAP database, the software filter is a non-SAP software filter, and the special database is a non-SAP database.

Detail Description Paragraph:

[0060] FIG. 8 is a flow chart 100 of DES software (called "DES UPDATE" software) for updating an execution document 32 of FIG. 2, in accordance with embodiments of the present invention. The execution document 32 includes, inter alia, a purchase order (see FIG. 5), a scheduling agreement (see FIG. 6), etc. As an example of updating a purchase order, the existing price is changed to a new price in accordance with a new or renewed contract (or for any other reason), the purchase order will be modified to incorporate the price change. The new price typically replaces the existing price in the purchase order not before the new price becomes effective for the contract. As an example of updating a scheduling agreement in which the price paid by the purchaser is the price in effect at the time of delivery of purchase item(s), then the scheduling agreement will be updated to reflect any change in price that occurs prior to delivery.

Detail Description Paragraph:

[0061] Generally, an "attribute value" is updated in an execution agreement in accordance with the present invention. An attribute value in an execution document is a contract parameter value in the execution document. Examples of attribute values include, inter alia, a price of a purchase item, delivery terms (e.g., F.O.B., F.A.S.), financing terms, etc.

Detail Description Paragraph:

[0062] As seen in FIGS. 2-3, the execution document 32 is derived from the contract dataset 28 of the contract datasets 42 of the DES RDBS 40. The execution document 32 may have been generated in a sequence described by blocks 101-103 of FIG. 8. In block 101, the contract dataset 28 is shown to originate in the PCMS 12 (see FIG. 1). Block 102 shows the contract dataset 28 transferred from the PCMS 12 to the DES 14 (see FIG. 1) as discussed supra in conjunction with the block 64 of FIG. 7. Alternatively, the contract dataset 28 may have been placed or generated in the DES 14 in any other manner such as from a process sequence of adding a new purchase item to the RDBS 40 of the DES 14 as described supra in conjunction with the process sequence starting with block 78 of FIG. 7. Block 103 in FIG. 8 depicts generation of the execution agreement in the DES 14 such that the execution agreement has an existing attribute value. Block 104 of FIG. 8 indicates that the DES 14 receives notice from the PCMS 12 that a new attribute value is now effective; i.e., that the contract dataset 28 has been modified to include the new attribute value for the associated purchase item. Accordingly, block 105 replaces the existing attribute value in the execution document with the new attribute value.

Detail Description Paragraph:

[0064] FIG. 9 depicts archiving a contract, in accordance with embodiments of the present invention. FIG. 9 shows a contract management architecture 120 comprising a PCMS 125 and X DES's, namely DES.sub.1, DES.sub.2, . . . , DES.sub.X wherein X.gtoreq.1. The contract management relationships between the PCMS 12 and DES 14 of FIG. 1, as described supra, apply to the PCMS 125 and DES.sub.1, DES.sub.2, . . . , DES.sub.X of FIG. 9. The PCMS 125 desires to archive (i.e., delete or store elsewhere) I contract datasets (I.gtoreq.1). Before actually implementing the archiving, the PCMS 125 requires unanimous approval of the archiving from all of DES.sub.1, DES.sub.2, . . . , DES.sub.X for each contract dataset to be archived. Accordingly, the PCMS 125 sends a list L of I identifiers to each of DES.sub.1, DES.sub.2, . . . , DES.sub.X. Each of the I identifiers identifies a contract dataset in the PCMS 125 earmarked by the PCMS 125 for archiving. After receiving the list L, each of DES.sub.1, DES.sub.2, . . . , DES.sub.X responds to the PCMS 125 by sending a return list R.sub.1, R.sub.2, . . . , R.sub.X, respectively. The return lists are DES-specific; i.e., R.sub.1, R.sub.2, . . . , R.sub.X are independent of one another and are specific to DES.sub.1, DES.sub.2, . . . , DES.sub.X, respectively. The return list sent by DES.sub.i includes M.sub.i of the I identifiers (0.ltoreq.M.sub.i.ltoreq.I) for i=1, 2, . . . , I. Each contract dataset identified in the return list of DES.sub.i is approved by DES.sub.i for archiving, for i=1, 2, . . . , I.

Detail Description Paragraph:

[0065] After receiving all of the return lists, the PCMS 125 generates an intersection list from R.sub.1, R.sub.2, . . . , R.sub.X. The intersection list is a logical intersection of R.sub.1, R.sub.2, . . . , R.sub.X; i.e., the intersection list contains those contract datasets that are common to each of R.sub.1, R.sub.2, . . . , R.sub.X. Accordingly, each contract dataset on the intersection list appears on each return list R.sub.1, R.sub.2, . . . , R.sub.X. The PCMS 125 archives each contract dataset appearing on the intersection list. Note that the intersection list may be empty (i.e., have no contract datasets therein). If the intersection list is empty, then no contract datasets are archived.

Detail Description Paragraph:

[0066] The PCMS 125 has software (called "PCMS ARCHIVE" software) for implementing FIG. 9; i.e.,: for preparing and sending the list L to each of DES.sub.1, DES.sub.2, . . . , DES.sub.X, for receiving R.sub.1, R.sub.2, . . . , R.sub.X and generating the intersection list, and for archiving the contract datasets appearing on the intersection list. Although the PCMS ARCHIVE software of the present invention does not currently exist in SAP, the scope of the present invention includes the PCMS ARCHIVE software as either SAP software or non-SAP software. In relation to use of the PCMS ARCHIVE software, the scope of the present invention includes the PCMS 12 and the DES 14 as each independently being a SAP system or a non-SAP system.

Detail Description Paragraph:

[0068] FIG. 10 is a block diagram of a computer configuration for the contract management architecture of FIG. 1 and the systems, databases, software, etc, of FIGS. 2-9, in accordance with embodiments of the present invention. FIG. 10 illustrates a computer network 80 comprising a PCMS 85 and a DES 95. The PMS 85 and the DES 95 communicate over a data path 88 such as communications network described supra in conjunction with the data path 15 of FIG. 1. The DES 95 represents one or more of such DES's which are linked to the PCMS 85.

CLAIMS:

1. A method for managing contract data, comprising: receiving a contract dataset by a decentralized execution system (DES) from a procurement contract management system (PCMS); and passing the contract dataset through a software filter that determines whether to store the contract dataset or a first portion thereof in a relational database of the DES, said relational database including contract datasets, vendor datasets, and purchase item datasets.
2. The method of claim 1, wherein the software filter further determines whether to store the contract dataset or a second portion thereof in a special database of the DES.
3. A method for managing contract data, comprising: receiving a contract dataset by a first SAP contract management system from a second SAP contract management system; and passing the contract dataset through a software filter that determines whether to store the contract dataset or a first portion thereof in a SAP database of the first SAP contract management system.
4. The method of claim 1, wherein the software filter further determines whether to store the contract dataset or a second portion thereof in a non-SAP database of the first SAP contract management system.
5. A method for managing contract data, comprising: receiving a contract datagroup D.sub.G by a decentralized execution system (DES) from a procurement contract management system (PCMS), said contract datagroup D.sub.G selected from the group consisting of a contract dataset and a contract deltadataset, said contract

datagroup D.sub.G identifying N purchase items purchasable from a vendor V keyed to the contract datagroup D.sub.G, said N at least 1, said contract datagroup D.sub.G identifying the vendor V if the contract datagroup D.sub.G is the contract dataset, said DES comprising a relational database that includes contract datasets, vendor datasets having vendors, and purchase item datasets having purchase items; determining which, if any, of the N purchase items identified in the contract datagroup D.sub.G match a purchase item in the purchase item datasets and determining a total number K of such purchase items in D.sub.G that do not so match a purchase item in the purchase item datasets, said K satisfying $0 < K \leq N$; and if $K < N$ then if the contract datagroup D.sub.G is the contract dataset then determining whether the vendor V matches a vendor in the vendor datasets and if the vendor V so matches a vendor in the vendor datasets then adding a subset of D.sub.G to the relational database, said subset of D.sub.G excluding the K purchase items from D.sub.G, else if the contract datagroup D.sub.G is the contract deltadataset and D.sub.G is keyed to a first contract dataset in the relational database then adding to the first contract dataset in the relational database a remaining $N-K$ purchase items of D.sub.G.

6. The method of claim 5, wherein the DES further comprises a special database that includes contract datasets, wherein the contract datagroup D.sub.G is the contract deltadataset, and wherein if $K > 0$ then said method further comprising: if D.sub.G is keyed to a first contract dataset in the special database, then adding to the first contract dataset in the special database the K purchase items of D.sub.G; and if D.sub.G is not keyed to any contract dataset in the special database, then forming from D.sub.G a contract dataset D.sub.C1 that includes the K purchase items and excludes the remaining $N-K$ purchase items, and adding D.sub.C1 to the special database.

7. The method of claim 5, wherein if $K < N$ and the contract datagroup D.sub.G is the contract dataset and the vendor V does not match a vendor in the vendor datasets, then further comprising adding a vendor dataset D.sub.V to the relational database when a contract based on the subset of D.sub.G is required at the DES, said vendor dataset D.sub.V keyed to the vendor V.

9. The method of claim 7, wherein adding D.sub.V to the relational database comprises: communicating a message to a DES buyer keyed to at least one purchase item of the remaining $N-K$ purchase items, each of said at least one purchase item matching a purchase item in the purchase item datasets, said message relating to adding D.sub.V to the relational database; and having the DES buyer cause D.sub.V to be added to the relational database when the contract based on the subset of D.sub.G is required at the DES.

10. The method of claim 5, wherein the contract datagroup D.sub.G is the contract dataset.

11. The method of claim 5, wherein the contract datagroup D.sub.G is the contract deltadataset.

13. A method for managing contract data, comprising: receiving a contract dataset D.sub.C by a decentralized execution system (DES) from a procurement contract management system (PCMS), said contract dataset D.sub.C identifying a vendor V and N purchase items purchasable from the vendor V, said N at least 1, said DES comprising a relational database that includes contract datasets, vendor datasets having vendors, and purchase item datasets having purchase items, said DES further comprising a special database that includes contract datasets; determining which, if any, of the N purchase items identified in the contract dataset D.sub.C match a purchase item in the purchase item datasets and determining a total number K of such purchase items in D.sub.C that do not so match a purchase item in the purchase item datasets, said K satisfying $0 < K \leq N$; and if $K = N$ then adding D.sub.C to the special database, else if $K < N$ then determining whether the vendor V

matches a vendor in the vendor datasets and if the vendor V so matches a vendor in the vendor datasets then adding a first subset of D.sub.C to the relational database and if $K > 0$ adding a second subset of D.sub.C to the contract datasets of the special database, said first subset of D.sub.C excluding the K purchase items from D.sub.C, said second subset of D.sub.C excluding a remaining $N-K$ purchase items from D.sub.C.

14. The method of claim 13, further comprising: adding a new purchase item to the purchase item datasets; determining whether the new purchase item is identified in a contract dataset D.sub.CS of the special database; and if the new purchase item is so identified in D.sub.CS and D.sub.CS identifies J purchase items such that J is at least 1, then determining whether a vendor identified in D.sub.CS matches a vendor in the vendor datasets and if the vendor identified in D.sub.CS so matches a vendor in the vendor datasets then: if a contract identifier of D.sub.CS matches a contract identifier of a first contract dataset in the relational database then adding the new purchase item to the first contract dataset, else if the contract identifier of D.sub.CS does not matches a contract identifier of any contract dataset in the relational database then adding a subset of D.sub.CS to the relational database, said subset of D.sub.CS including the new purchase item; and if $J=1$ then deleting D.sub.CS from the special database else deleting the new purchase item from D.sub.CS.

17. A method for updating an execution document relating to a contract, said method comprising: having an execution document at a decentralized execution system (DES) of a procurement contract management system (PCMS), said execution document being derived from a contract dataset in the DES, said execution document having an existing attribute value for a purchase item in the contract dataset; receiving notice at the DES from the PCMS of a new attribute value that is to replace the existing attribute value; and replacing the existing attribute value with the new attribute value in the execution document.

22. A method of contract archiving, comprising: sending a list of I identifiers by a procurement contract management system (PCMS) to at least one decentralized execution system (DES), said I at least 1, each identifier of the I identifiers identifying a contract dataset in the PCMS earmarked by the PCMS for archiving; receiving by the PCMS a return list of M of the I identifiers from each DES of the at least one DES in response to said sending, said M in a range of $0.1 \text{ to } \text{req.M.} \text{ to } \text{req.I}$, said return list being DES-specific, each said contract dataset identified in the return list of each DES having been approved by said each DES for archiving; and archiving by the PCMS each contract dataset identified in the list of I identifiers and appearing in an intersection list of the return lists, if the intersection list is not empty.

23. The method of claim 22, further comprising communicating by the PCMS to each DES of the at least one DES: that the archiving was done by the PCMS for the contract datasets appearing in the intersect list, if the intersection list is not empty; or that the archiving will not be done, if the intersection list is empty.

25. A method of contract archiving, comprising: receiving by a first decentralized execution system (DES) of at least one DES from a procurement contract management system (PCMS) a list of I identifiers, said I at least 1, each identifier of the I identifiers identifying a contract dataset in the PCMS earmarked by the PCMS for archiving, said list of I identifiers sent by the PCMS to each DES of the at least one DES, said PCMS adapted to receive a return list of M of the I identifiers from each DES of the at least one DES in response to said sending, said M in a range of $0.1 \text{ to } \text{req.M.} \text{ to } \text{req.I}$, said return list being DES-specific, each said contract dataset identified in the return list of each DES having been approved by said each DES for archiving, said PCMS adapted to archive each contract dataset identified in the suggest list and appearing in an intersection list of the return lists if the intersection list is not empty; and sending by the first DES to the PCMS the return

list of the first DES.

26. The method of claim 25, further comprising receiving by the first DES notification from the PCMS: that the archiving was done by the PCMS for the contract datasets appearing in the intersect list, if the intersection list is not empty; or that the archiving will not be done, if the intersection list is empty.

28. A system for managing contract data, comprising software at a decentralized execution system (DES), said software adapted to: receive a contract dataset by the DES from a procurement contract management system (PCMS); and pass the contract dataset through a software filter that is adapted to determine whether to store the contract dataset or a first portion thereof in a relational database of the DES, said relational database adapted to include contract datasets, vendor datasets, and purchase item datasets.

29. The system for managing contract data of claim 28, wherein the software filter is adapted to further determine whether to store the contract dataset or a second portion thereof in a special database of the DES.

30. A system for managing contract data, comprising software at a decentralized execution system (DES), said software adapted to: receive a contract dataset by a first SAP contract management system from a second SAP contract management system; and pass the contract dataset through a software filter that determines whether to store the contract dataset or a first portion thereof in a SAP database of the DES.

31. The system for managing contract data of claim 30, wherein the software filter is adapted to further determine whether to store the contract dataset or a second portion thereof in a non-SAP database of the first SAP contract management system.

32. A system for managing contract data, comprising software at a decentralized execution system (DES), said software adapted: to have the DES receive a contract datagroup D.sub.G from a procurement contract management system (PCMS), said contract datagroup D.sub.G selected from the group consisting of a contract dataset and a contract deltadataset, said contract datagroup D.sub.G identifying N purchase items purchasable from a vendor V keyed to the contract datagroup D.sub.G, said N at least 1, said contract datagroup D.sub.G identifying the vendor V if the contract datagroup D.sub.G is the contract dataset, said DES comprising a relational database that includes contract datasets, vendor datasets having vendors, and purchase item datasets having purchase items; to determine which, if any, of the N purchase items identified in the contract datagroup D.sub.G match a purchase item in the purchase item datasets and to determine a total number K of such purchase items in the D.sub.G that do not so match a purchase item in the purchase item datasets, said K satisfying $0 < K \leq N$; and if $K < N$ then if the contract datagroup D.sub.G is the contract dataset then to determine whether the vendor V matches a vendor in the vendor datasets and if the vendor V so matches a vendor in the vendor datasets then to add a subset of D.sub.G to the relational database, said subset of D.sub.G excluding the K purchase items from D.sub.G, else if the contract datagroup D.sub.G is the contract deltadataset and said contract deltadataset is keyed to a first dataset in the relational database then to add to the first dataset a remaining $N - K$ purchase items of the contract datagroup D.sub.G.

33. The system for managing contract data of claim 32, wherein the DES further comprises a special database that includes contract datasets, wherein the contract datagroup D.sub.G is the contract deltadataset, and wherein if $K > 0$ then said software is further adapted: if D.sub.G is keyed to a first contract dataset in the special database, then to add to the first contract dataset in the special database the K purchase items of D.sub.G; and if D.sub.G is not keyed to any contract dataset in the special database, then to form from D.sub.G a contract dataset

D.sub.C1 that includes the K purchase items and excludes the remaining N-K purchase items, and to add D.sub.C1 to the special database.

34. The system for managing contract data of claim 32, wherein if $K < N$ and the contract datagroup D.sub.G is the contract dataset and the vendor V does not match a vendor in the vendor datasets, then said software is further adapted to have a vendor dataset D.sub.V added to the relational database when a contract based on the subset of D.sub.G is required at the DES, said vendor dataset D.sub.V keyed to the vendor V.

35. The system for managing contract data of claim 34, wherein said software is further adapted to have the vendor dataset D.sub.V extracted from a vendor database prior to having D.sub.V added to the relational database.

36. The system for managing contract data of claim 34, wherein to have the vendor dataset D.sub.V added to the relational database comprises: to communicate a message to a DES buyer keyed to at least one purchase item of the remaining N-K purchase items, each of said at least one purchase item matching a purchase item in the purchase item datasets, said message relating to adding D.sub.V to the relational database; and to have the DES buyer cause D.sub.V to be added to the relational database when the contract based on the subset of D.sub.G is required at the DES.

37. The system for managing contract data of claim 32, wherein the contract datagroup D.sub.G is the contract dataset.

38. The system for managing contract data of claim 32, wherein the contract datagroup D.sub.G is the contract deltadataset.

39. The system for managing contract data of claim 32, said PCMS being a SAP system, said DES being a SAP system, said relational database being a SAP database, said software being non-SAP software.

40. A system for managing contract data, comprising software at a decentralized execution system (DES), said software adapted: to have the DES receive a contract dataset D.sub.C from a procurement contract management system (PCMS), said contract dataset D.sub.C identifying a vendor V and M purchase items purchasable from the vendor V, said M at least 1, said DES comprising a relational database that includes contract datasets, vendor datasets having vendors, and purchase item datasets having purchase items, said DES further comprising a special database that includes contract datasets; to determine which, if any, of the N purchase items identified in the contract dataset D.sub.C match a purchase item in the purchase item datasets and to determine a total number K of such purchase items in the D.sub.C that do not so match a purchase item in the purchase item datasets, said K satisfying $0 < K \leq N$; and if $K = N$ then to add D.sub.C to the special database, else if $K < N$ then to determine whether the vendor V matches a vendor in the vendor datasets and if the vendor V so matches a vendor in the vendor datasets then to add a first subset of D.sub.C to the relational database and if $K > 0$ to add a second subset of D.sub.C to the contract datasets of the special database, said first subset of D.sub.C excluding the K purchase items from D.sub.C, said second subset of D.sub.C excluding a remaining N-K purchase items from D.sub.C.

41. The system for managing contract data of claim 40, wherein said software is further adapted: to add a new purchase item to the purchase item datasets; to determine whether the new purchase item is identified in a contract dataset D.sub.CS of the special database; and if the new purchase item is so identified in D.sub.CS and D.sub.CS identifies J purchase items such that J is at least 1, then to determine whether a vendor identified in D.sub.CS matches a vendor in the vendor datasets, and if the vendor identified in D.sub.CS so matches a vendor in the vendor datasets then: if a contract identifier of D.sub.CS matches a contract

identifier of a first contract dataset in the relational database then to add the new purchase item to the first contract dataset, else if the contract identifier of D.sub.CS does not matches a contract identifier of any contract dataset in the relational database then to add a subset of D.sub.CS to the relational database, said subset of D.sub.CS including the new purchase item; and J=1 then to delete D.sub.CS from the special database else to delete the new purchase item from D.sub.CS.

42. The system for managing contract data of claim 41, wherein said software is further adapted to extract the new purchase item from a purchase item database prior to adding the new purchase item to the purchase item datasets.

43. The system for managing contract data of claim 40, said PCMS being a SAP system, said DES being a SAP system, said relational database being a SAP database, said special database being a non-SAP database, said software being non-SAP software.

44. A system for updating an execution document relating to a contract, comprising a decentralized execution system (DES) of a procurement contract management system (PCMS), said DES having software adapted: to have an execution document at the DES, said execution document being derived from a contract dataset in the DES, said execution document having an existing attribute value for a purchase item in the contract dataset; to receive notice at the DES from the PCMS of a new attribute value that is to replace the existing attribute value; and to replace the existing attribute value with the new attribute value in the execution document.

49. A system for contract archiving, comprising a procurement contract management system (PCMS) having software adapted: to send a list of I identifiers to at least one decentralized execution system (DES), said I at least 1, each identifier of the I identifiers identifying a contract dataset in the PCMS earmarked by the PCMS for archiving; to receive a return list of M of the I identifiers from each DES of the at least one DES in response to having sent the list of I identifiers to each said DES, said M in a range of $0.<M \leq I$, said return list being DES-specific, each said contract dataset identified in the return list of each DES having been approved by said each DES for archiving; and to archive each contract dataset identified in the list of I identifiers and appearing in an intersection list of the return lists, if the intersection list is not empty.

50. The system for contract archiving of claim 49, said software further adapted to communicate to each DES of the at least one DES: that the archiving was done by the PCMS for the contract datasets appearing in the intersect list, if the intersection list is not empty; or that the archiving will not be done, if the intersection list is empty.

51. The system for contract archiving of claim 49, said PCMS and each of the at least one DES being a SAP system, said software being non-SAP software.

52. A system for contract archiving, comprising a first decentralized execution system (DES) of at least one DES, said first DES having software adapted: to receive from a procurement contract management system (PCMS) a list of I identifiers, said I at least 1, each identifier of the I identifiers adapted to identify a contract dataset in the PCMS earmarked by the PCMS for archiving, said list of I identifiers adapted to be sent by the PCMS to each DES of the at least one DES, said PCMS adapted to receive a return list of M of the I identifiers from each DES of the at least one DES in response to having sent the list of I identifiers to each said DES, said M in a range of $0.<M \leq I$, said return list being DES-specific, each said contract dataset identified in the return list of each DES having been approved by each said DES for archiving, said PCMS adapted to archive each contract dataset identified in the list of I identifiers and appearing in an intersection list of the return lists if the intersection list is

not empty; and to send to the PCMS the return list of the first DES.

53. The system for contract archiving of claim 52, said software further adapted to receive notification from the PCMS: that the archiving was done by the PCMS for the contract datasets appearing in the intersect list, if the intersection list is not empty; or that the archiving will not be done, if the intersection list is empty.

54. The system for contract archiving of claim 52, said PCMS and each of the at least one DES being a SAP system, said software being non-SAP software.

First Hit

Generate Collection

Print

L5: Entry 4 of 11

File: PGPB

May 1, 2003

DOCUMENT-IDENTIFIER: US 20030083897 A1

TITLE: Contract management aid

Abstract Paragraph:

(a) a contract information database configured to receive data from a computerised contract management system, said database including data on each contract, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and (iii) the dependency, if any known, of the contract on any other of the contracts; and

Summary of Invention Paragraph:

[0001] The present invention relates to the field of contract management systems to facilitate contract fulfilment and provides a system to facilitate management of a plurality of inter-related contracts. This may suitably be incorporated into or used alongside an existing computerised contract management system.

Summary of Invention Paragraph:

[0005] According to a first aspect of the present invention there is provided apparatus for facilitating management of a plurality of inter-related contracts, which apparatus comprises: (a) a contract information database configured to receive data from a computerised contract management system, said database including data on each contract, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and (iii) the dependency, if any known, of the contract on any other of the contracts; and (b) a processor operatively linked to the database and configured to process the data on each contract from the contract information database and use the known dependency data from the database to compute the dependency of each contract on any of the other contracts to generate a representative structure having nodes and linkages between nodes representative of the inter-relationship of the plurality of inter-related contracts.

Summary of Invention Paragraph:

[0012] The apparatus of the present invention may further comprise an adaptor interface for receiving the data from the computerised contract management system and to re-format the data to a format standardised for storage in the contract information database. This adaptor interface may suitably include adaptors for a number of different existing contract information databases and may be implemented in hardware or software.

Summary of Invention Paragraph:

[0013] The apparatus is preferably further configured to poll the associated computerised contract management system to obtain required data for storage by the contract information database.

Summary of Invention Paragraph:

[0014] Particularly preferably the system is further configured to poll the computerised contract management system throughout the performance of the contract to obtain update information on the current status of at least one of and preferably all of the plurality of inter-related contracts. Conversely, where the computerised contract management system is configured to be integral with or part

of the apparatus of the present invention, the apparatus for facilitating contract management may itself be configured to automatically transmit update contract status information to the contract information database.

Summary of Invention Paragraph:

[0015] According to a second aspect of the present invention there is provided a processor for facilitating management of a plurality of inter-related contracts, operable to receive data on each contract of the plurality of inter-related contracts from a computerised contract management system, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and (iii) the dependency, if any known, of the contract on any other of the contracts; and further operable to process the data on each contract and use the known dependency data to compute the dependency of each contract on any of the other contracts to generate a structure having nodes and linkages between nodes representative of the inter-relationship of the plurality of inter-relating contracts.

Summary of Invention Paragraph:

[0016] According to a third aspect of the present invention there is provided a method of facilitating management of a plurality of inter-related contracts, said method comprising: receiving data on each contract of the plurality of inter-related contracts from a computerised contract management system, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and (iii) the dependency, if any known, of the contract on any other of the contracts; and processing the data on each contract and using the known dependency data to generate a structure having nodes and linkages between nodes representative of the inter-relationship of the plurality of inter-relating contracts.

Summary of Invention Paragraph:

[0017] According to a fourth aspect of the present invention there is provided a computer program operable to execute a method of facilitating management of a plurality of inter-related contracts, said method comprising: receiving data on each contract of the plurality of inter-related contracts from a computerised contract management system, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and (iii) the dependency, if any known, of the contract on any other of the contracts; and processing the data on each contract and using the known dependency data to generate a structure having nodes and linkages between nodes representative of the inter-relationship of the plurality of inter-relating contracts.

Summary of Invention Paragraph:

[0018] According to a fifth aspect of the present invention there is provided a computer program stored on a datacarrier and operable to execute a method of facilitating management of a plurality of inter-related contracts, said method comprising: receiving data on each contract of the plurality of inter-related contracts from a computerised contract management system, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and (iii) the dependency, if any known, of the contract on any other of the contracts; and processing the data on each contract and using the known dependency data to generate a structure having nodes and linkages between nodes representative of the inter-relationship of the plurality of inter-relating contracts.

Detail Description Paragraph:

[0029] Ideally, all this information is provided from a computerised contract management system 140. However, most such systems 140 do not contain all necessary information about dependencies and therefore a user may be expected to input such details himself. A data entry interface 110 is accordingly suitably provided linked to the contract information database 100 for the input of these details by the user. This may, of course, comprise a computer with keyboard and/or mouse and VDU

that may be the same as or integral with the graphing processor 120 and display screen 160. Suitably a GUI is displayed on the display screen 160 and is configured for simplified manual entry of the relevant data by the user and having suitable prompts for the data required.

Detail Description Paragraph:

[0030] There are presently a substantial number of different computerised contract management systems 140 used commercially, many of which have their own distinctive format and programmatic interface and, accordingly, the present system is provided with an adaptor interface 130 to be able to interact with a particular computerised system and produce information about a contract in a standard format (e.g. defined by an XML schema). The adaptor interface 130 performs a format conversion taking information in one structure and placing it into a different format; containing empty fields where necessary. If a (single) standard form for the information exists this layer becomes unnecessary. For instances where the system needs to pull information from a number of computerised contract management systems 140 it will have a respective adaptor within the adaptor interface 130 for each different system 140.

Detail Description Paragraph:

[0032] Once the system receives data from the contract management system 140 it will try to match it against its existing data, forming a type of graph structure linking the data into a form that can be reasoned about as a structure. For this there is suitably consistency between the contract ID numbers; although there may be a mapping mechanism in the adaptors to translate contract IDs into a standard form.

Detail Description Paragraph:

[0074] The database now contains a set of entries which collectively define the intended graph structure containing information about contracts and their stages and their relationships with other contracts. Where placeholder nodes have been created the user should be prompted that no information exists about dependent contracts such that they can enter what information they have or tell the system which contract management system contains details and therefore to download this information in a similar manner.

Detail Description Paragraph:

[0094] On viewing the display illustrated in FIG. 4 it becomes obvious that the first stage of the final contract, as represented by highlighted node 440, will fail. By then looking more closely at the problem information the user may establish that it is due to a verification report, or due to problem events as received from the contract management system. Details may be obtained by selecting a node and, for example, selecting from a menu of types of information available.

Detail Description Paragraph:

[0107] Since the graph can become very crowded where large volumes of data are represented, filters may be applied and the user can focus in on areas of the time-line and then scroll the line.

CLAIMS:

1. Apparatus for facilitating management of a plurality of inter-related contracts, which apparatus comprises: (a) a contract information database configured to receive data from a computerised contract management system, said database including data on each contract, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and (iii) the dependency, if any known, of the contract on any other of the contracts; and (b) a processor operatively linked to the database and configured to process the data on each contract from the contract information database and use the known dependency data from the database to compute the dependency of each contract on any of the other

contracts to generate a representative structure having nodes and linkages between nodes representative of the inter-relationship of the plurality of inter-related contracts.

9. Apparatus as claimed in claim 1, further comprising an adaptor interface for receiving data from the computerised contract management system and to re-format the data to a format standardised for storage in the contract information database.

10. Apparatus as claimed in claim 9, further configured to poll the associated computerised contract management system to obtain required data for storage by the contract information database.

11. Apparatus as claimed in claim 9, further configured to poll the computerised contract management system throughout the performance of the contract to obtain update information on the current status of at least one of the plurality of inter-related contracts.

12. Apparatus as claimed in claim 1, wherein the computerised contract management system is configured to be integral with or part of the apparatus for facilitating contract management, and the apparatus for facilitating contract management is configured to automatically transmit update contract status information to the contract information database.

13. A processor for facilitating management of a plurality of inter-related contracts, operable to receive data on each contract of the plurality of inter-related contracts from a computerised contract management system, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and (iii) the dependency, if any known, of the contract on any other of the contracts; and further operable to process the data on each contract and use the known dependency data to compute the dependency of each contract on any of the other contracts to generate a structure having nodes and linkages between nodes representative of the inter-relationship of the plurality of inter-relating contracts.

14. A method of facilitating management of a plurality of inter-related contracts, said method comprising: receiving data on each contract of the plurality of inter-related contracts from a computerised contract management system, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and (iii) the dependency, if any known, of the contract on any other of the contracts; and processing the data on each contract and using the known dependency data to generate a structure having nodes and linkages between nodes representative of the inter-relationship of the plurality of inter-relating contracts.

15. A computer program operable to execute a method of facilitating management of a plurality of inter-related contracts, said method comprising: receiving data on each contract of the plurality of inter-related contracts from a computerised contract management system, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and (iii) the dependency, if any known, of the contract on any other of the contracts; and processing the data on each contract and using the known dependency data to generate a structure having nodes and linkages between nodes representative of the inter-relationship of the plurality of inter-relating contracts.

16. A computer program stored on a data carrier and operable to execute a method of facilitating management of a plurality of inter-related contracts, said method comprising: receiving data on each contract of the plurality of inter-related contracts from a computerised contract management system, the data comprising at least: (i) the identity of the contract; (ii) a deadline for the contract; and

- (iii) the dependency, if any known, of the contract on any other of the contracts; and processing the data on each contract and using the known dependency data to generate a structure having nodes and linkages between nodes representative of the inter-relationship of the plurality of inter-relating contracts.

Print Request Result(s)

Printer Name: cpk2_4c32_gbfrptr

Printer Location: cpk2__4c32

Number of Copies Printed : 1

- US20030074277: Ok
- US20030041049: Ok
- US20030083897: Ok
- US20020091539: Ok

OK

Back to List